

Application Number 10/675,909
Response to Office Action mailed February 26, 2008

REMARKS

This amendment is responsive to the Office Action dated February 26, 2008. Applicant has amended claims 1-10, 12-14, cancelled claim 11, and added claim 21. Claims 1-10 and 12-21 are pending upon entry of this amendment.

Claim Rejection Under 35 U.S.C. § 102

In the Office Action, the Examiner rejected claims 1-20 under 35 U.S.C. 102(e) as being anticipated by Elkin et al. (U.S. Patent Publication No. 2007/0179828). Applicant respectfully traverses the rejection to the extent applicable to Applicant's amended claims. Elkin et al. fails to disclose each and every feature of the claimed invention, as required by 35 U.S.C. 102(b), and provides no teaching that would have suggested the desirability of modification to include such features.

Claim 1

Applicant has amended claim 1 to recite an administration console that supports node-level modification of the enterprise planning model individual nodes of the model while the application server executes the enterprise planning session in accordance with the model. Further, amended claim 1 requires that, during the enterprise planning session, the administration console allows an analyst to check-out individual nodes of the model for editing without taking the model offline and without preventing execution of the enterprise planning session by the application server in accordance with the model. In this way, Applicant has amended claim 1 to clarify that the administration console allows nodes of an enterprise planning model to be individually updated while an enterprise planning session is being executed in accordance with that same model. Support for the claim amendments can be found, for example, within paragraphs [051] and [052] of the present application.

Elkin et al. fails to teach or suggest a system in which an administration console supports node-level modification of individual nodes of the enterprise planning model while the application server executes the enterprise planning session in accordance with the model. Moreover, Elkin et al. fails to teach or suggest a system in which the administration console allows an analyst to check-out individual nodes of the model for editing during execution of the

Application Number 10/675,909
Response to Office Action mailed February 26, 2008

enterprise planning session without taking the model offline and without preventing execution of the enterprise planning session by the application server in accordance with the model, as further required by claim 1.

For example, in part, the Examiner relied on Elkin at ¶ [0013], which states:

[0013] The present invention meets these goals by incorporating a set of software tools that allow the graphical definition of top-down workflow process models. Once defined, these models are completely useable enterprise applications that can be deployed in real-time without interrupting current business operations.

Thus, ¶ [0013] states only that after models are defined (i.e., “once defined”), the models can be deployed as enterprise software applications “without interrupting current business operations.” This suggests at best that other business operations are not interrupted.

In relevant part, Elkin at ¶ [0195] describes deployment of process model onto the process servers:

[0195] When the process model 100 has been defined, the process designer 300 generates a deployment package and installs it on a process server 500. The deployment package contains all the necessary information to execute the run time application, including the compiled process model 100, related classes and objects, and middleware adapters 240. The deployment package also verifies the consistency and completeness of process 120 definitions, and the check-in status of repository objects 312.

As stated in Elkin at ¶ [0195], to deploy a process model, the entire module must be “compiled” to produce a deployment package that can then be installed on a process server. Elkin makes clear that the deployment package includes all the necessary information to execute the run time application including the compiled process model, related classes and objects, and middleware adapters.

With respect to updating a process model after deployment and during execution, Elkin at ¶ [0196] states:

[0196] The installation of an updated process model deployment package can be carried out while the servers 500 are up and running. This mechanism allows overlaying an updated or a new process model 100 on the running servers 500 in real-time. While an updated process model 100 is being deployed, tasks 130 already in progress can be carried out according the old definition of the task 130.

Application Number 10/675,909
Response to Office Action mailed February 26, 2008

As evidence by the above paragraph, Elkin states that an updated process model deployment package (i.e., the package described in ¶ [0196] of Elkin as containing all necessary information for execution including the entire model) can be overlapped with an existing model currently being executed. This fails to teach a system that supports node-level modification of individual nodes of the enterprise planning model while the application server executes the enterprise planning session in accordance with the model, as required by amended claim 1. That is, Elkin specifically requires the modification, compilation and redeployment of an entire "deployment package." Elkin provides no mechanism that enables individual nodes to be modified and redeployed. For this reason, Elkin et al. fails to teach or suggest a system in which the administration console allows an analyst to check-out individual nodes of the model for editing during execution of the enterprise planning session without taking the model offline and without preventing execution of the enterprise planning session by the application server in accordance with the model, as further required by claim 1.

For at least these reasons, Elkin fails to establish a prima facie case for anticipation of Applicant's claims 1 under 35 U.S.C. 102(e). Withdrawal of this rejection is requested.

Independent claims 7 and 14

As discussed above, Elkin requires the modification, compilation and redeployment of a deployment package that includes the model itself, and does not teach a mechanism that enables nodes to be modified and redeployed individually. Thus, for similar reasons, Elkin fails to teach or suggest a method comprising executing an enterprise planning session in accordance with an enterprise planning model, wherein the enterprise planning model defines hierarchically arranged nodes associated with business logic software modules and enterprise contributors, checking-out an individual one of the nodes of the model for editing during execution of the enterprise planning session in accordance with the enterprise planning model, and modifying the checked-out node of the model without preventing execution of the enterprise planning session for the nodes of the enterprise planning model that are not checked-out, as required by claim 7.

Similarly, Elkin fails to teach or suggest a computer-readable storage medium comprising instructions that cause a processor to check-out an individual one of the nodes of the model for editing during execution of the enterprise planning session in accordance with the enterprise

Application Number 10/675,909
Response to Office Action mailed February 26, 2008

planning model, and modify the checked-out node of the model without preventing execution of the enterprise planning session for the nodes of the enterprise planning model that are not checked out, as required by claim 14.

Withdrawal of these rejections is requested.

Dependent claims 2 and 8

Elkin fails to teach or suggest a system in which an administration console receives updated model information from the analyst for the checked-out nodes, and updates a respective slice of the enterprise planning model for only the checked-out nodes based on the updated model information, as required by claim 2. As described above, Elkin requires the modification, compilation and redeployment of a deployment package that includes the model and all other information necessary to deploy and run the model. Elkin does not teach a mechanism that enables an analyst to update, during execution of an enterprise planning session in accordance with an enterprise planning model, only a slice of that model for an individual node within that model.

For similar reasons, Elkin fails to teach or suggest a method in which modifying the checked-out node comprises: receiving updated model information for the checked-out nodes, and updating a respective slice of the enterprise planning model for only the checked-out nodes based on the updated model information. Withdrawal of these rejections is requested.

Application Number 10/675,909
Response to Office Action mailed February 26, 2008

Dependent claims 4 and 10

Elkin fails to teach or suggest a system in which the administration console reconciles the contribution data that was received during the enterprise planning session prior to the check-out of the individual nodes with updated model information when the checked-out nodes are subsequently checked-in during the execution of the enterprise planning session, as required by claim 4.

As described above, Elkin requires the modification, compilation and redeployment of a deployment package that includes the model and all other information necessary to deploy and run the model. Elkin does not teach any mechanism for reconciling contribution data that was received during the enterprise planning session prior to the check-out of the individual nodes with updated model information *when the checked-out nodes are subsequently checked-in during the execution of the enterprise planning session*. As described above, Elkin fails to provide any mechanism that allows individual nodes to be checked-out, modified, and checked-in during the execution of the planning session. Consequently, Elkin provides no mechanism for reconciliation of the data with the updated model information when the checked-out nodes are subsequently checked-in during the execution of the enterprise planning session, as required by amended claim 4.

With respect to these elements, the Examiner cites Elkin at paragraphs [0196]–[1097]. However, in relevant part, these paragraphs state: “While an updated process model 100 is being deployed, tasks 130 already in progress can be carried out according the old definition of the task 130.” This makes clear that Elkin requires that tasks in progress are “carried out” according to the old definition of the task. Thus, Elkin avoids the need to reconcile data for any given task (node) when a node is updated because, as clearly stated by Elkin, tasks in progress must execute to completion in accordance with the old model definition. In contrast, Applicant’s claim 4 requires reconciliation of data that was received prior to check-out for individual nodes when those nodes are subsequently checked-in. This effectively allows nodes of a model to be modified even though existing users may already be executing the planning session with respect to those nodes. See, for example, paragraph [051] of the pending application that describes reconciliation of even open user sessions with respect to model changes when those changes are put into production. Elkin does not provide or suggest such a mechanism.

Application Number 10/675,909
Response to Office Action mailed February 26, 2008

For similar reason, Elkin fails to teach or suggest a method that requires reconciling the contribution data that was received prior to the check-out of the individual nodes with the updated model information when the checked-out nodes are subsequently checked-in during the execution of the enterprise planning session. Withdrawal of these rejections is requested.

Dependent claims 6 and 13

Applicant's claim 6 recites a particular mechanism for reconciling data for an updated node. Specifically, claim 6 requires that administration console *pushes reconciliation jobs to the local computing devices of the enterprise contributors for execution on the local computing devices to reconcile the previously received contribution data entered by the enterprise contributors with the updated model information*. As explained in paragraph [051] of the pending application, this approach may be advantageous in that enterprise planning system need not be taken offline to update enterprise models, and that computing resources to process the updates can be distributed across the remote computing devices of users. Elkin provides no teaching or suggestion of a mechanism that pushes reconciliation jobs the local computing devices of the users participating within an enterprise planning session. Elkin describes only a process server 500 that executes the model and any updated version of that model.

For similar reasons, Elkin fails to teach or suggest a method that requires defining reconciliation jobs for execution by remote computers of the enterprise contributors to reconcile the previously received contribution data with the updated model information for the checked-in nodes, as required by claim 13. Withdrawal of these rejections is requested.

Application Number 10/675,909
Response to Office Action mailed February 26, 2008

CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date:

By:

May 16, 2008
SHUMAKER & SIEFFERT, P.A.
1625 Radio Drive, Suite 300
Woodbury, Minnesota 55125
Telephone: 651.735.1100
Facsimile: 651.735.1102

Kent J. Sieffert
Name: Kent J. Sieffert
Reg. No.: 41,312